PROJECT REPORT

NEWS TRACKER APPLICATION

# INTRODUCTION

* 1. **Project Overview**

The main objective of the project is to provide people a web application through which people can access all types of news and information.Through this application,any user can gain technical knowledge of the world and its surrounding with just one click ahead.

User does not have to visit multiple sites for different related information. All information is going to be in one place.

# Purpose

The purpose is to develop an web application, which will

eliminate the problems faced in the current scenario. This

application will provide all the information and news related to E-

sports, sciennce, and Technology or that are in trend at one place.

So, it will save time and efforts of the users by making it more

efficient Using, this application will terminate the possibility of

information redundancy.

# LITERATURE SURVEY

* 1. **Existing Problem**

Our research utilizes the current trends and technology to design and develop a web application for every user. News organizations whose web applications only provide users with their articles or videos are missing a big opportunity. There are multiple news sharing apps used by a single user and are often spammed with notifications.There is also a lot of fake news.

# Reference

1. Exploring mobile news reading interactions for news app personalisation – Marios Constantinde & John Dowell.
2. Detection and Tracking in News Articles – Sagar Patel, Sanket suthar.
3. Following the fed with a news tracker – Michael william Mc Cracken

# Problem Statement

The customer needs a way to get relevant news based on his choices so that the customer does not have to spend a lot of time on searching news.News is filled with ads and spams annoys and irritates the customer and affects the user experience.Market is full of news with all categories. Customers are not interested in all the categories and will be more interested with their personal choices.Traditional way of tracking

news is slow and obsolete,

user needs a new innovative application to

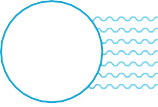
track news with personal based choices.

# IDEATION AND PROPOSED SOLUTION

* 1. **Empathy map canvas**



# Ideation & Brainstorming



* 1. **Proposed Solution**

As a team, we realized that we have different skill sets and felt that we would all be able to cater to this problem statement in a very

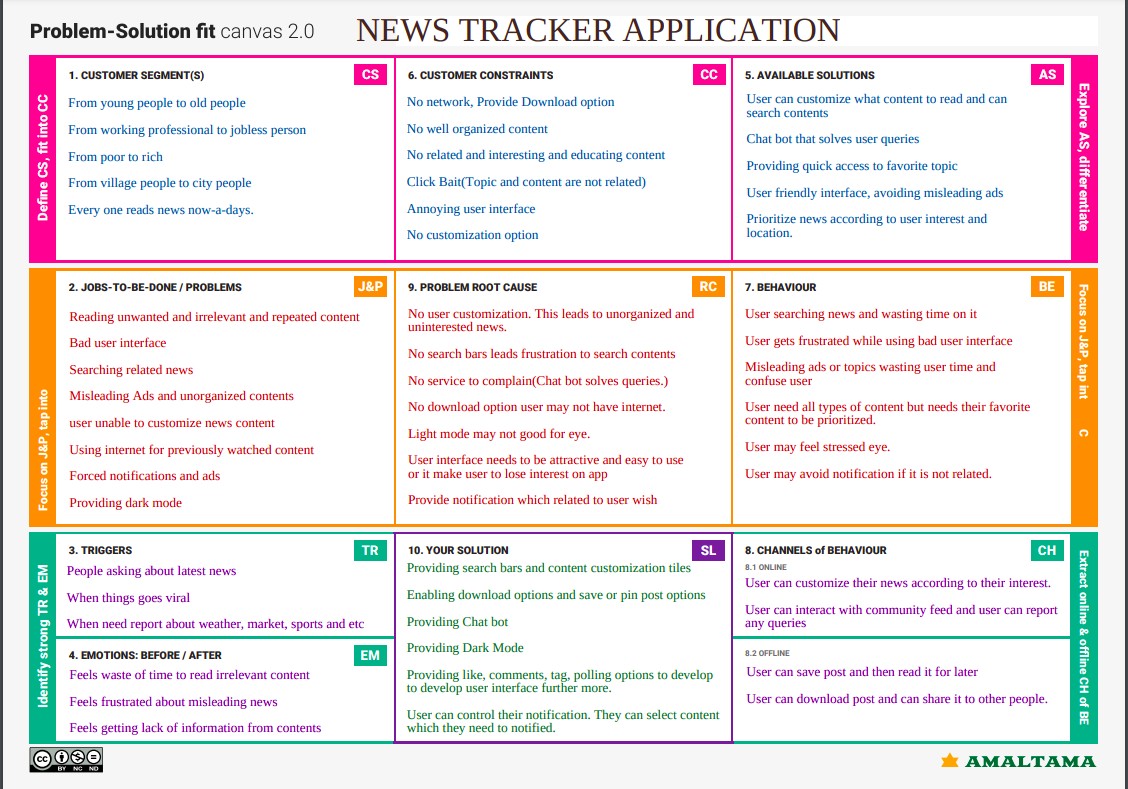
strategic and collaborative manner.

Plus it was interesting to see

how all of us had a very different take on news consumption. A

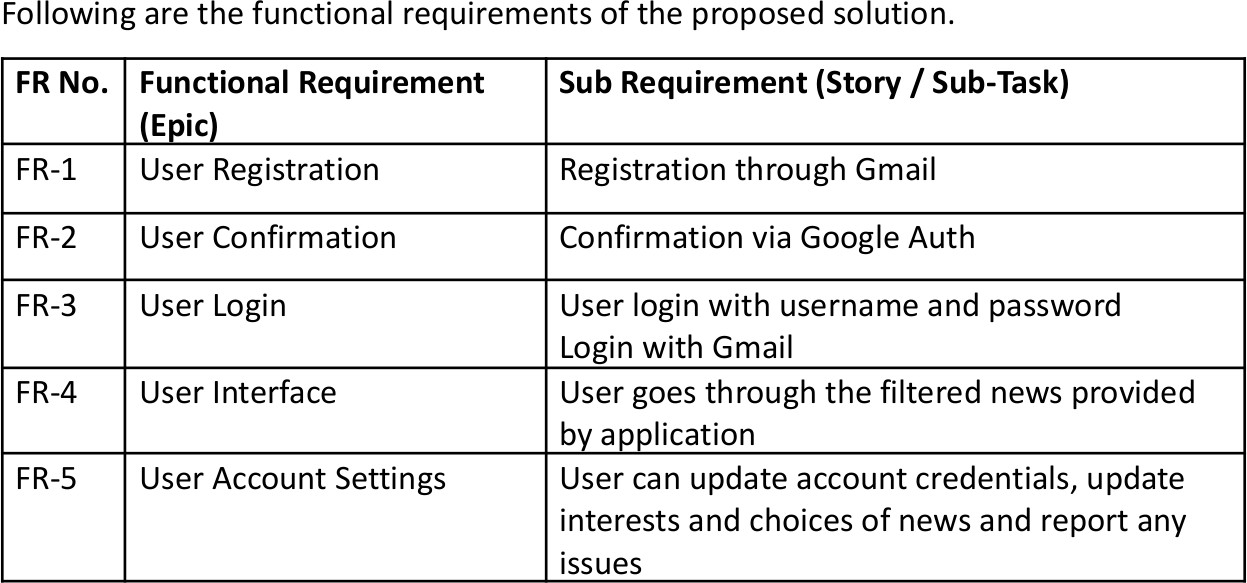
news-sharing app wants to help users find relevant and important news easily every day and also understand explicitly that the news is not fake but from proper sources

# Problem Solution fit

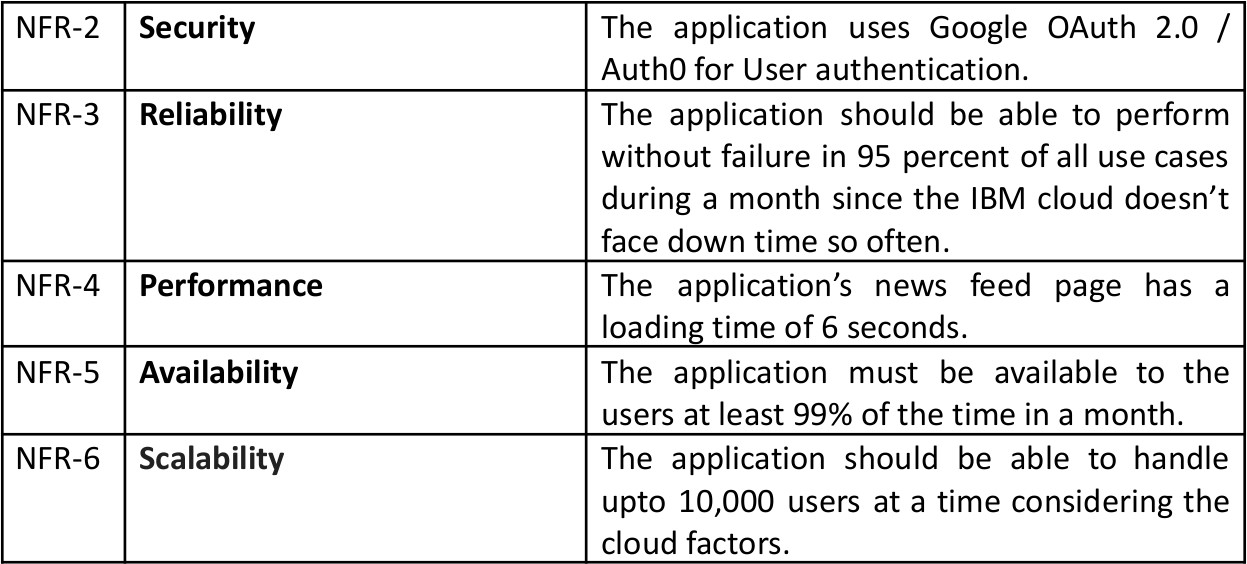


1. **REQUIREMENT ANALYSIS**

# Functional Requirement

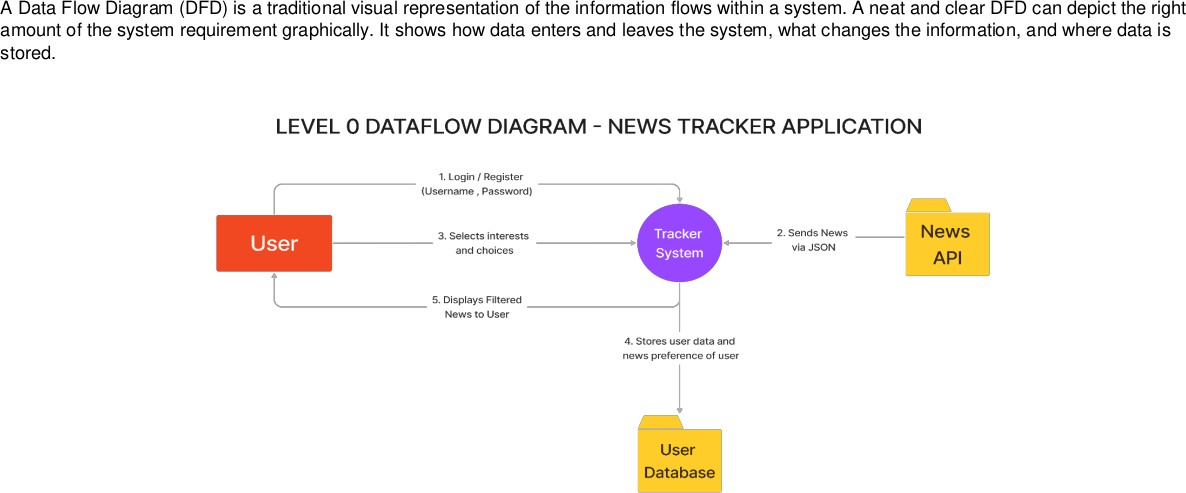


* 1. **Non-Functional Requirements**

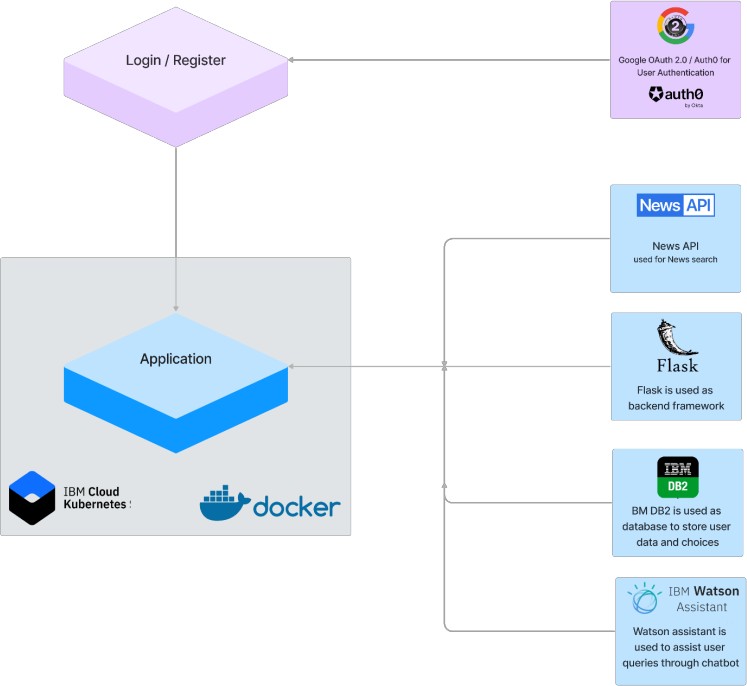


# PROJECT DESIGN

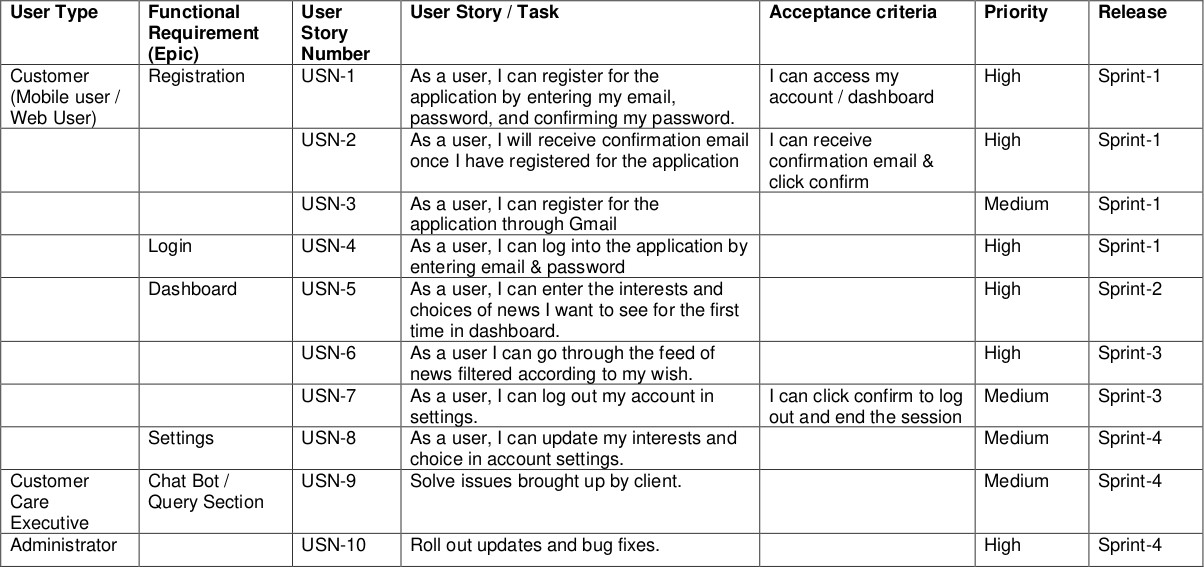
* 1. **Data Flow Diagrams**



# Solution & Technical Architecture

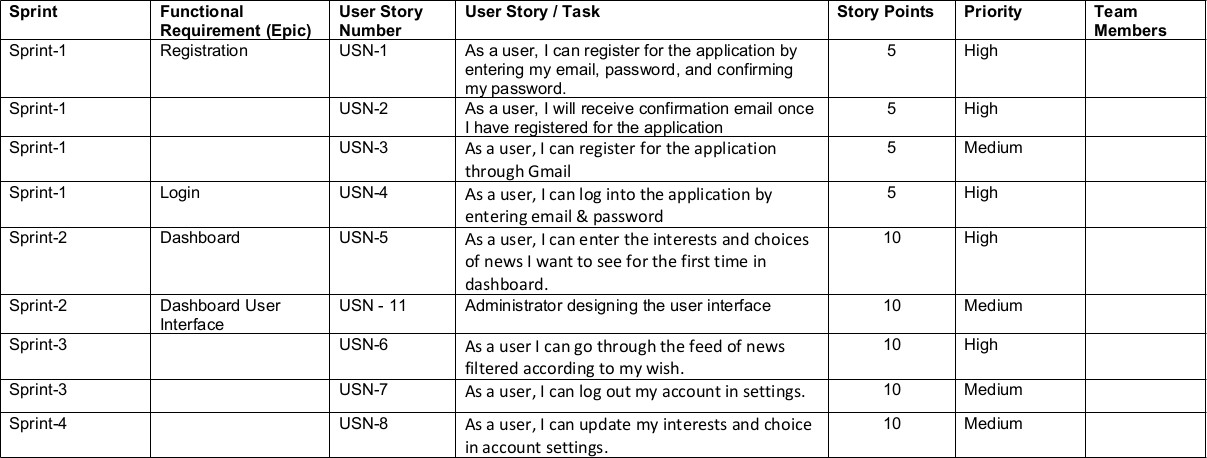


* 1. **User Stories**

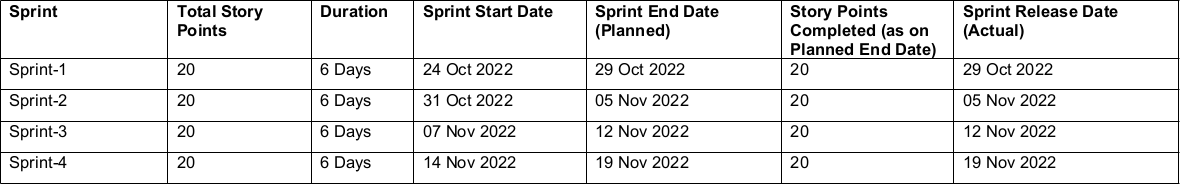


# PROJECT PLANNING & SCHEDULING

* 1. **Sprint Planning & Estimation**



# Sprint Delivery Schedule

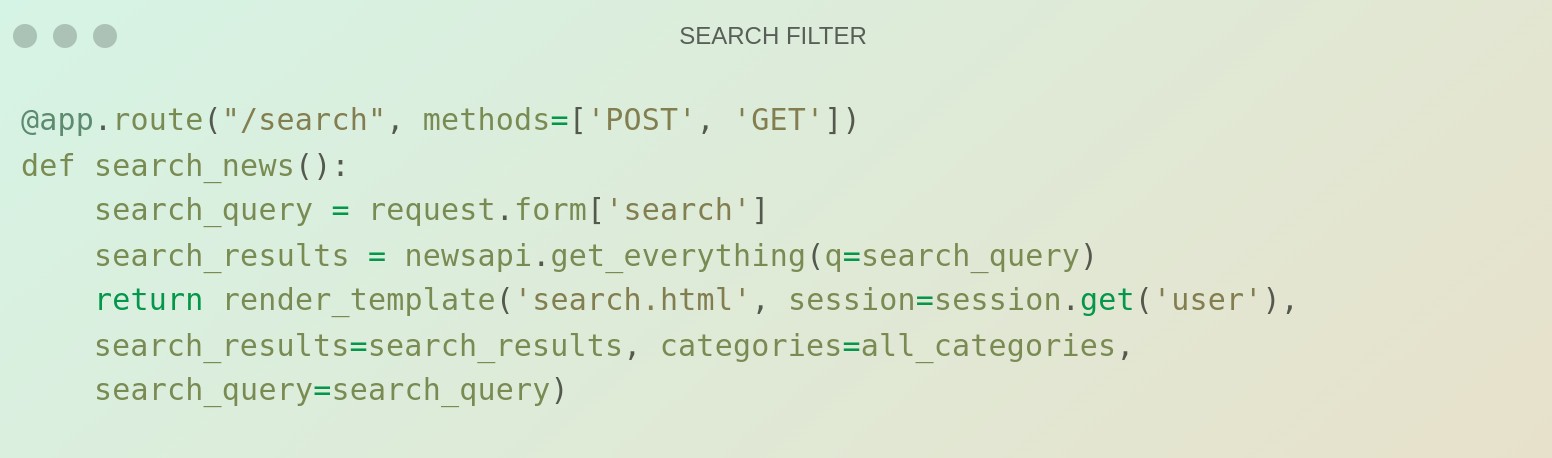


1. **CODING & SOLUTIONING**

# 7.1

**FEATURE 1 –**

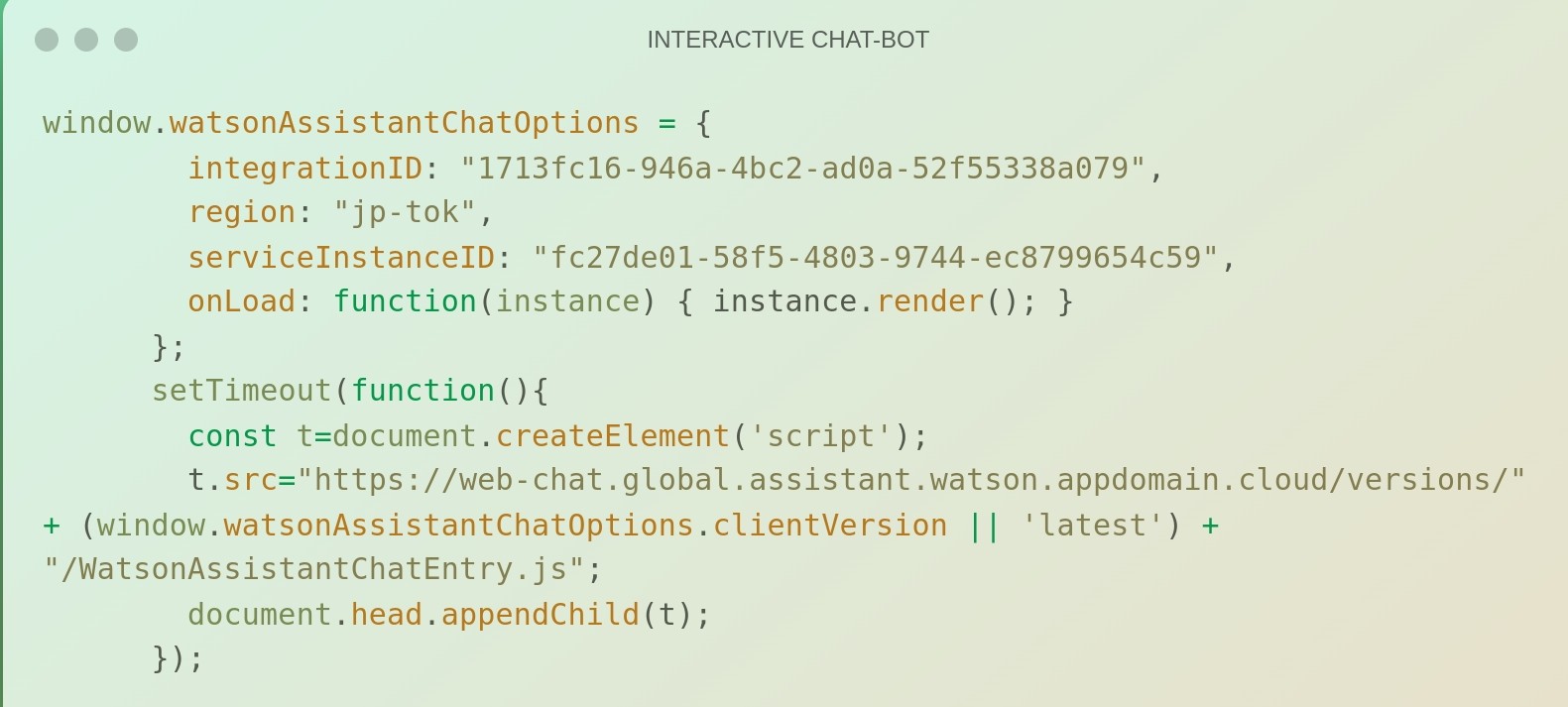
# Search Filter



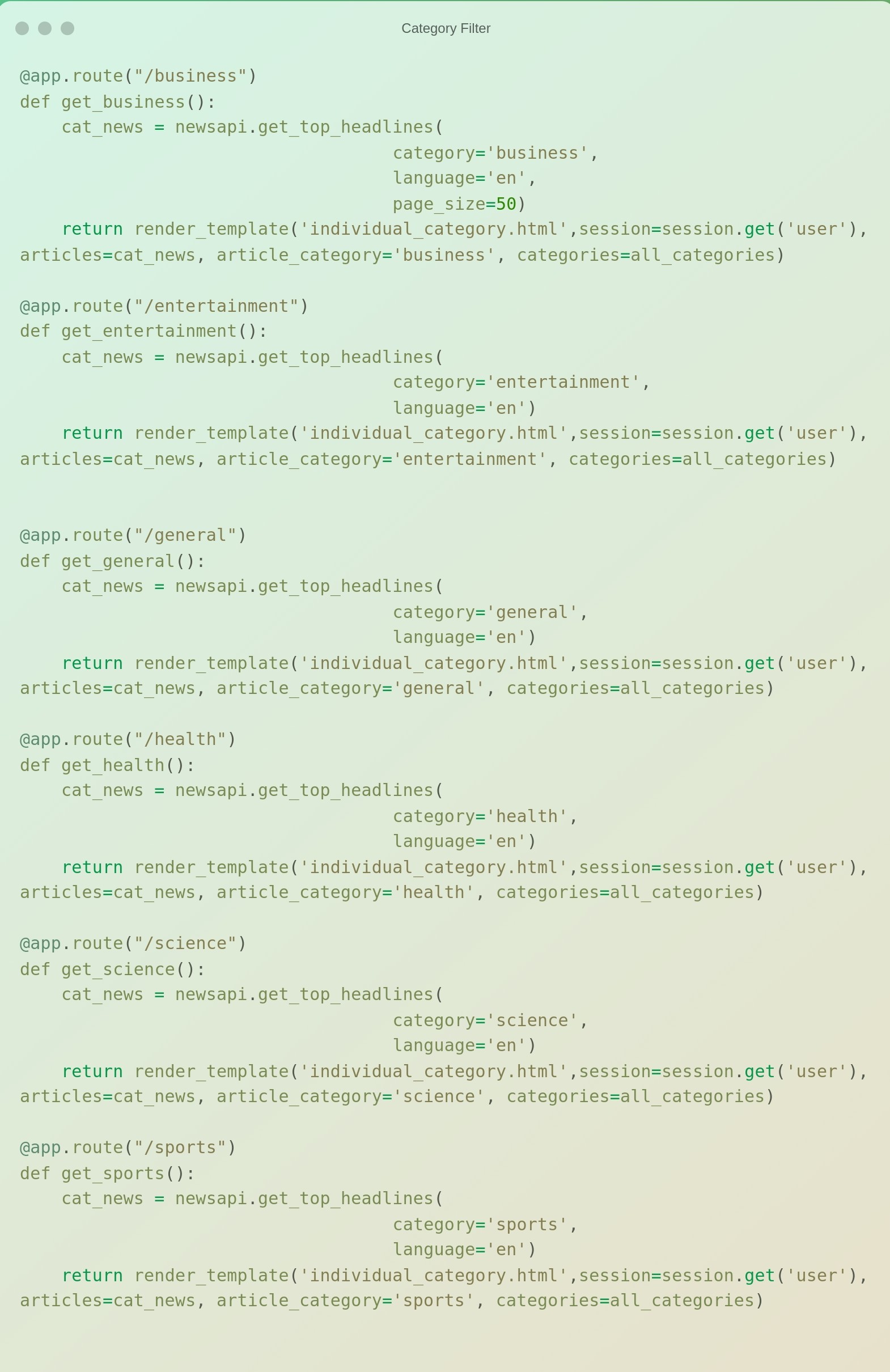
**7.2**

# FEATURE 2 -

**Chat-bot**



# 7.3 Category Filter



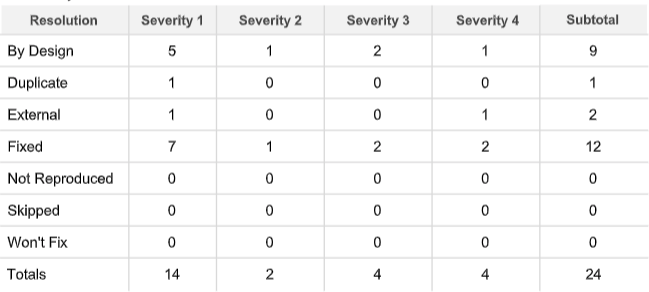
* 1. **User Acceptance Testing**

# PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the News Tracker Application project at the time of the release to User Acceptance Testing(UAT).

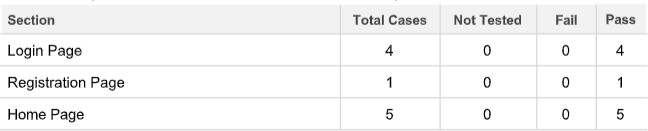
# DEFECT ANALYSIS

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.



# TEST CASE ANALYSIS

This report shows the number of test cases that have passed, failed, and untested .



# 9. ADVANTAGES

1. Viewers can get their news straight off their smartphone or tablet computer.
2. News is at their fingertips in an instant. An online newspaper can

be read more elaborate than a printed newspaper.

1. You can read the old issues too very easily at the click of the mouse.
2. This will help the users to share news on various platforms such as

Twitter and Facebook. experience

# DISADVANTAGES

This will not only give an amazing user

1. It CAN BE limited by time.
2. It may rely too heavily on personalities, emotions, opinions… not facts. 3.It can shortchange complex stories or avoid them altogether.

# CONCLUSION

The Motivation and scope behind this project are to connect people through this application and provide a medium to share their views on the topic/news/information. Then, People with the same interest can interact with each other. However, they can even share more information on the topic. This app while cross-checks the redundancy in the information along with the false and misleading information,which later results in panic in the people.

# FUTURE SCOPE

Location feature with automation can be implemented which means as user move from one city to other local news will change as per it. Offline Reading can be improve will more efficient way on full articles. Data quality check needed. If API can’t reach to certain article source it gives null value which can cause problem in JSON parsing .

# APPENDIX SOURCE CODE

import json

from os import environ as env

from urllib.parse import quote\_plus, urlencode

from authlib.integrations.flask\_client import OAuth from dotenv import find\_dotenv, load\_dotenv

from flask import Flask, redirect, render\_template, session, url\_for, request

from newsapi import NewsApiClient

ENV\_FILE = find\_dotenv() if ENV\_FILE:

load\_dotenv(ENV\_FILE)

app = Flask( name )

# env\_config = env.getenv("APP\_SETTINGS", "config.DevelopmentConfig")

# app.config.from\_object(env\_config) app.secret\_key = env.get("APP\_SECRET\_KEY")

newsapi = NewsApiClient(api\_key=env.get("NEWS\_API\_KEY"))

oauth = OAuth(app)

oauth.register( "auth0",

client\_id= env.get("AUTH0\_CLIENT\_ID"), client\_secret= env.get("AUTH0\_CLIENT\_SECRET"), client\_kwargs={

"scope": "openid profile email",

},

server\_metadata\_url=f'https://{env.get("AUTH0\_DOMAIN")}/.we ll-known/openid-configuration'

)

all\_categories = ["business", "entertainment", "general", "health", "science", "sports", "technology"]

# @app.route("/") # def index():

# sample\_config\_data = app.config.get("SAMPLE") # return sample\_config\_data

@app.route("/login")

def login():

return oauth.auth0.authorize\_redirect( #should be redirect\_uri and not redirect\_url

redirect\_uri=url\_for("callback", \_external=True)

)

@app.route("/callback", methods=["GET", "POST"])

def callback():

token = oauth.auth0.authorize\_access\_token() session["user"] = token

return redirect("/")

@app.route("/logout")

def logout(): session.clear() return redirect(

"https://" + env.get("AUTH0\_DOMAIN")

+ "/v2/logout?"

+ urlencode(

{

"returnTo": url\_for("home", \_external=True), "client\_id": env.get("AUTH0\_CLIENT\_ID"),

},

quote\_via=quote\_plus,

)

)

@app.route("/")

def home():

# all\_articles = get\_news()

all\_articles = get\_top\_headlines\_today() return render\_template(

# "home.html", "home\_og.html", session=session.get('user'), all\_articles = all\_articles, categories= all\_categories

# pretty=json.dumps(session.get('user'),indent=4)

)

@app.route("/news")

def get\_news():

all\_articles = newsapi.get\_everything(q='bitcoin',

sources='bbc-news,the-verge', domains='bbc.co.uk,techcrunch.com', from\_param='2022-10-10',

to='2022-10-12',

language='en', sort\_by='relevancy', page=1)

return all\_articles

# @app.route("/ind-cat", methods=["POST", "GET"]) # def get\_individual\_category\_news():

# category\_news = newsapi.get\_everything(category)

@app.route("/top-headlines-today")

def get\_top\_headlines\_today():

user\_categories = ["business", "entertainment", "general", "health", "science", "sports", "technology"]

top\_headlines = {}

for i in range(len(user\_categories)): top\_headlines[user\_categories[i]] =

newsapi.get\_top\_headlines(

# sources="", category=user\_categories[i], language='en',

page\_size=5)

# top\_headlines.append(newsapi.get\_top\_headlines(

# #

return top\_headlines

category='business', language='en'))

@app.route("/user-info")

def user\_info():

return session.get('user')

@app.route("/news-from-sources")

def get\_news\_from\_sources(): return

@app.route("/search", methods=['POST', 'GET'])

def search\_news():

search\_query = request.form['search']

search\_results = newsapi.get\_everything(q=search\_query) # return search\_results

return render\_template('search.html', session=session.get('user'), search\_results=search\_results, categories=all\_categories, search\_query=search\_query)

@app.route("/business")

def get\_business():

cat\_news = newsapi.get\_top\_headlines(

category='business', language='en', page\_size=50)

return render\_template('individual\_category.html',session=session.get('u ser'), articles=cat\_news, article\_category='business', categories=all\_categories)

@app.route("/entertainment")

def get\_entertainment():

cat\_news = newsapi.get\_top\_headlines(

category='entertainment', language='en')

return render\_template('individual\_category.html',session=session.get('u ser'), articles=cat\_news, article\_category='entertainment', categories=all\_categories)

@app.route("/general")

def get\_general():

cat\_news = newsapi.get\_top\_headlines(

category='general', language='en')

# return cat\_news return

render\_template('individual\_category.html',session=session.get('u

ser'), articles=cat\_news, article\_category='general', categories=all\_categories)

@app.route("/health")

def get\_health():

cat\_news = newsapi.get\_top\_headlines(

category='health', language='en')

# return cat\_news return

render\_template('individual\_category.html',session=session.get('u

ser'), articles=cat\_news, article\_category='health', categories=all\_categories)

@app.route("/science")

def get\_science():

cat\_news = newsapi.get\_top\_headlines(

category='science', language='en')

# return cat\_news return

render\_template('individual\_category.html',session=session.get('u

ser'), articles=cat\_news, article\_category='science', categories=all\_categories)

@app.route("/sports")

def get\_sports():

cat\_news = newsapi.get\_top\_headlines(

category='sports', language='en')

# return render\_template(cat\_news) return

render\_template('individual\_category.html',session=session.get('u

ser'), articles=cat\_news, article\_category='sports', categories=all\_categories)

@app.route("/technology")

def get\_technology():

cat\_news = newsapi.get\_top\_headlines(

category='technology', language='en')

# return cat\_news return

render\_template('individual\_category.html',session=session.get('u

ser'), articles=cat\_news, article\_category='technology', categories=all\_categories)

if name == " main ":

app.run(host="0.0.0.0", port=env.get("PORT", 5000))

# GITHUB REPO LINK - [Source Code](https://github.com/IBM-EPBL/IBM-Project-29083-1660120673) [Click here to watch live](https://drive.google.com/file/d/1px2FhydT_X8sBxBbo2Vb95lGO29dnTu5/view?usp=sharing) .